# BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

## IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.**

# WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Thursday, July 29, 2004

| Hide? | Set Name | Query | Hit Count |
|---|---|---|---|
| | | *DB=USPT,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR* | |
| ☐ | L29 | l17 and L28 | 22 |
| ☐ | L28 | (bios with (resum$7 or restart$4)) same ((os or (operating adj system)) with (resum$7 or restart$4)) | 85 |
| ☐ | L27 | L25 same OS | 4 |
| ☐ | L26 | L25 same BIOS | 0 |
| ☐ | L25 | l24 with power$4 | 174 |
| ☐ | L24 | restor$9 with parallel$4 | 3493 |
| ☐ | L23 | l17 and L22 | 9 |
| ☐ | L22 | (restor$7 with (operating adj system)) same (restor$7 with devices) | 37 |
| ☐ | L21 | (restor$7 near5 (operating adj system)) same (restor$7 near5 devices) | 5 |
| ☐ | L20 | l1 and L17 | 2 |
| ☐ | L19 | l5 and L17 | 0 |
| ☐ | L18 | l4 and L17 | 13 |
| ☐ | L17 | l12 or l13 or l14 or l15 or L16 | 3096 |
| ☐ | L16 | 710/10.ccls. | 550 |
| ☐ | L15 | 713/300.ccls. | 863 |
| ☐ | L14 | 713/100.ccls. | 682 |
| ☐ | L13 | 713/2.ccls. | 852 |
| ☐ | L12 | 713/1.ccls. | 1010 |
| ☐ | L11 | (restor$4 near3 (operat$4 adj environment)) same (sleep$4 or powerdown or suspend) | 2 |
| ☐ | L10 | L4 and (BIOS same (operating adj system)) | 46 |
| ☐ | L9 | L8 and (operating adj system) | 64 |
| ☐ | L8 | l4 and BIOS | 123 |
| ☐ | L7 | l4 same BIOS | 13 |
| ☐ | L6 | (restor$4 with system with devices with parallel$4) | 7 |
| ☐ | L5 | (restor$4 same system same devices same parallel$4) | 102 |
| ☐ | L4 | (restor$4 same parallel$4) | 8879 |
| ☐ | L3 | (first adj resum$5) same (second adj resum$5) | 10 |
| ☐ | L2 | (first adj resume adj process) same (second adj resume adj process) | 0 |
| ☐ | L1 | (restor$4 near3 (operat$4 adj environment)) | 21 |

L11: Entry 1 of 2 File: USPT May 7, 2002

DOCUMENT-IDENTIFIER: US 6385721 B1
** See image for Certificate of Correction **
TITLE: Computer with bootable hibernation partition

Detailed Description Text (5):
Thus, a PC may have multiple mass storage devices attached to it to store several operating systems, programs, or data. However, to describe how to access the programs or data on the mass storage devices, the mass storage organization must be known. Physically, most mass storage devices are organized using the terminology of hard disk drives. Hard disks typically have at least one platter for storing programs or data. Each platter is divided into a number of concentric storage units called tracks. A track is further divided into sectors. Each platter is accessed by a top head and a bottom head which read and write data onto the hard disk. Logically, a hard disk may be divided into partitions, each partition having an amount of storage selectable at the time of creation of the partition, as long as the aggregate of all the partitions' storage area does not exceed the storage capacity of the hard disk. For example, a single 8 Mbyte hard disk could have three partitions logically named C: with 4 Mbytes of storage, D: with 3 Mbytes, and E: with 1 MByte. Partitions are further divisible into tracks, cylinders, and sectors for addressing purposes. Under the MS-DOS/Windows FDISK utility supplied by Microsoft, for example, a PC may have up to four disk partitions of differing types and sizes. The differing types may be MS-DOS/Windows types such as FAT12, FAT16, FAT32, or other non MS-DOS/Windows types, thus allowing more than one OS to be installed on the same hard disk. Also, additional disk partitions types are reserved for future uses. One of these reserved partitions has been designated to be used to store the contents of the PC memory during a sleep state of the PC. This sleep state is often referred to as "hibernation mode", or alternatively, "save-to-disk mode". Hibernation mode allows the PC to conserve power by being in an off state but still maintaining the operating environment of the user before entering this sleep state. When the PC is turned back on from hibernation mode, the PC is restored to previous operating environment thus reducing boot time. Typically, one partition on a mass storage device in the PC is used to store the PC's memory and this "hibernation partition" is made non-bootable and is hidden from the operating system so that it can not be accessed or corrupted. Since the control of the PC power states are typically handled by the BIOS, only the BIOS code is aware of how to access and read and write the data areas of a hibernation partition.

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L21: Entry 2 of 5 File: USPT Jun 5, 2001

DOCUMENT-IDENTIFIER: US 6243831 B1
TITLE: Computer system with power loss protection mechanism

Detailed Description Text (18):
The S4 sleeping state is the lowest power state because RAM is turned off, in contrast to the S1, S2, and S3 sleeping states, where RAM continues to be powered. In the S4 sleeping state, all device configurations are saved to RAM under the direction of the operating system. Then, the operating system saves RAM to a hibernation file somewhere in a non-volatile storage medium. The operating system then powers down the computer system S, except for trickle current, in some cases. Resume from the S4 state results from a variety of events, dependent upon designer choices. For example, an incoming fax or phone call may trigger a return from S4. When resume occurs, the contents of the hibernation file are restored to RAM by the operating system, and all devices are re-powered and reconfigured to the pre-S4 state.

Previous Doc Next Doc Go to Doc#

L21: Entry 4 of 5 File: USPT Aug 3, 1999

DOCUMENT-IDENTIFIER: US 5931954 A
** See image for Certificate of Correction **
TITLE: I/O control apparatus having check recovery function

Brief Summary Text (31):
Another object is to provide a software layer between an operating system kernel and an existing device driver which restores the state of the I/O devices when the computer system rolls back upon a fault.

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

**End of Result Set**

Generate Collection Print

L21: Entry 5 of 5 File: USPT May 31, 1994

DOCUMENT-IDENTIFIER: US 5317752 A
TITLE: Fault-tolerant computer system with auto-restart after power-fall

Detailed Description Text (121):
Device drivers during restart will now be considered. Device state is restored only for devices designated "resume on restart". If "reboot on restart" was chosen, of course, device state which was in existence during the shutdown will not be restored, as the system is rebooted with a fresh copy of the operating system.

Previous Doc Next Doc Go to Doc#

L23: Entry 4 of 9 File: USPT May 21, 2002

DOCUMENT-IDENTIFIER: US 6393584 B1
TITLE: Method and system for efficiently saving the operating state of a data processing system

Detailed Description Text (19):
If, however, the software hook detects a sleep file on hard disk drive 20, the process proceeds to block 131, which illustrates determining from the data returned by POST if the configuration of data processing system 10 has changed since the system state was stored. If devices have added to or removed from data processing system 10, the process proceeds to block 130, which has been described. Verifying that the configuration of data processing system 10 remains unchanged ensures that operating system settings for each device within data processing system 10 may be correctly restored. If the devices within data processing system 10 at restart are consistent with the devices present when the operating state was saved, the process proceeds to blocks 132-136, which illustrate restoring the operating state of data processing system 10 from the sleep file on hard disk drive 20. First, at block 132 virtual memory pages that are classified as resident are restored to RAM 46. Additional pages of nonresident data may also be paged from the swap file if the performance of hard disk drive 20 is sufficient to avoid an unacceptable user-perceived delay in restoring the operating state. Next, the hardware configurations of other components of data processing system 10 are restored at blocks 134 and 136. Blocks 134 and 136 illustrate the restoration of hardware configurations in a preferred embodiment of the present invention in which at least some of the components of data processing system 10 meet the Advanced Power Management (APM) specifications. At block 134, the hardware configurations of devices that do not support APM are restored from the sleep file. Since devices which support APM can recall their prior operating system settings, at block 136 the hardware configurations of APM-equipped devices are restored by transmitting an APM message from microprocessor 30 to these devices. Upon the restoration of the operating state of data processing system 10, the user will perceive that data processing system 10 is in the same state that it was in prior to invoking sleep state. For example, returning to FIG. 4, display 22 will again display window 102, graphical pointer 104, and icons 106-110. Thereafter, the process depicted in FIG. 5 terminates at block 130.

Current US Cross Reference Classification (1):
713/100

Current US Cross Reference Classification (2):
713/2

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#
Generate Collection Print

L23: Entry 1 of 9 File: USPT May 4, 2004

DOCUMENT-IDENTIFIER: US 6732280 B1
TITLE: Computer system performing machine specific tasks before going to a low power state

Brief Summary Text (10):
The ACPI specification required certain hardware components and extensions to facilitate the hardware interface. Typically an embedded microcontroller was one of these required hardware components. This embedded microcontroller was in effect a stand-alone processor in each ACPI-compliant peripheral device that worked hand-in-hand with the main processor of the computer system. One of the other additional hardware extensions necessary under the ACPI specification has been that of general purpose power-management registers. The power-management registers controlled the power state of the computer system. An ACPI-compatible operating system running on the processor of the computer system wrote instructions defining a power state for at least some of these registers. The ACPI compliant operating system controlled a variety of system-wide features, such as enabling and disabling interrupt sources, controlling the amount of power provided to various buses and devices, and restoring the computer system from low power states, such as sleep-mode and soft-off mode. For example, if the operating system desired to put the computer system in a sleep mode, the operating system wrote a "prepare to sleep" command to the registers. The operating system then issued an enable signal for certain of the registers, which caused an appropriate enable bit to be set in the registers. After the "prepare to sleep" instruction was sent and the registers were enabled, the computer system transitioned to whatever low-power state the operating system commanded.

Current US Original Classification (1):
713/300

Previous Doc Next Doc Go to Doc#

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#
Generate Collection Print

L23: Entry 2 of 9 File: USPT Mar 11, 2003

DOCUMENT-IDENTIFIER: US 6532535 B1
TITLE: Method for managing primary and secondary storage devices in an intelligent backup and restoring system

Detailed Description Text (3):
In the following description, the intelligent backup and restore system will be referred to by its development name "Genesis," however, emphasis will be placed on describing the technical functionality and software interactions between its various components, a host computer system's operating system, and peripheral storage devices and associated media.

Current US Original Classification (1):
713/1

Current US Cross Reference Classification (1):
713/2

Previous Doc Next Doc Go to Doc#

L23: Entry 6 of 9 File: USPT Nov 30, 1999

DOCUMENT-IDENTIFIER: US 5995454 A
TITLE: Computer system with alarm power-on function and automatic starting method thereof

Detailed Description Text (33):
If the power supply has not been turned on by the alarm (if the power supply has been turned on by the power switch operation), a usual power-on process will be carried out and the automatic starting process for starting the system operation will be started (step 4-3). The system operation includes supplying electric power to all of the devices and booting up the operating system (or the resume process of restoring the information saved immediately before the power supply was turned off).

Detailed Description Text (35):
If the "year, month, day-of-month" for automatic start-up coincides with the present "year, month, dayof-month," the usual power-on process will be carried out as when the power supply has been turned on by the power switch operation. That is, the automatic starting process to start the system operation will be started (step 4-3). The system operation includes supplying electric power to all of the devices and booting up the operating system (or the resume process of restoring the information saved immediately before the power supply was turned off).

Current US Cross Reference Classification (2):
713/300

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L23: Entry 8 of 9 File: USPT Dec 2, 1997

DOCUMENT-IDENTIFIER: US 5694583 A
TITLE: BIOS emulation parameter preservation across computer bootstrapping

Current US Cross Reference Classification (1):
713/2

CLAIMS:

4. For a data processing system of a type comprising a central processing unit (CPU), a system memory for storing data in electrical signal form, a first port for receiving an input device generating electrical input signals and at least one second port for supplying electrical output signals to output devices, a non-volatile mass storage medium storing multiple independent applications in the form of data on respective BIOS emulated images, a drive for the non-volatile mass storage medium, and a system bus interconnecting the CPU, system memory, first and second parts and the drive for the mass storage medium:

a BIOS for carrying out prescribed functions including converting operating signals developed by an operating system executed by the CPU into electrical signals compatible with devices that are responsive to other signals supplied by the CPU to the system bus, the BIOS including an extended BIOS data area containing media emulation parameters in emulation tables and including programming defining a protected volatile memory region of said system memory and responsive to a command for a warm boot for duplicating and preserving only across the warm boot the media emulation parameters in the protected volatile memory region, initiating an operating system reboot sequence including clearing regions of the system memory other than the protected volatile memory region, and restoring the new media emulation parameters into the emulation tables in the extended BIOS data area.

Previous Doc Next Doc Go to Doc#

L29: Entry 1 of 22 File: USPT Mar 2, 2004

DOCUMENT-IDENTIFIER: US 6701273 B2
TITLE: Method and apparatus for controlling internal heat generating circuitry

Detailed Description Text (91):
More specifically, the system BIOS saves a system status (e.g., the contents of the register of the CPU 111 or various I/O registers) necessary for resuming the operating system or an application program which is being executed in a main memory 113 and also stores a suspend flag representing a suspend state in the backed-up CMOS memory of a real-time clock 120, and a HOT-INS flag representing that the portable computer 1 is docked in a power ON state (hot insertion) in a predetermined memory of the deskstation interface 117. The system BIOS issues a dock power OFF command to the power supply controller 123 (step C11). The dock power OFF command is a command for designating to temporarily power off the portable computer 1 for docking and set a suspend state and is sent to the portable computer 1 through the communication register of the deskstation interface 117.

Detailed Description Text (113):
More specifically, the system BIOS saves a system status (e.g., the contents of the register of the CPU 111 and various I/O registers) necessary for resuming the operating system or an application program which is being executed in the main memory 113 and also stores a suspend flag representing a suspend state for undocking in the CMOS memory of the real-time clock 120. The system BIOS issues an eject power ON command to the power supply controller 123 (step D7).

Current US Cross Reference Classification (1):
713/300

L29: Entry 2 of 22 File: USPT Feb 10, 2004

DOCUMENT-IDENTIFIER: US 6691234 B1
** See image for Certificate of Correction **
TITLE: Method and apparatus for executing instructions loaded into a reserved portion of system memory for transitioning a computer system from a first power state to a second power state

Brief Summary Text (7):
In the ACPI S3 sleeping state, all system context is lost except system memory. Hardware maintains system memory context by placing the system memory in a self-refresh mode when power is removed from the system. One of the responsibilities of system BIOS during a resume or transition from an S3 sleeping state is to reinitialize the system and transfer control to the operating system (OS) waking vector without corrupting the OS image. This places an awkward requirement on system BIOS, which is to execute code without using the system memory. This requirement prevents the use of a stack, which is needed in order to develop resume code with a high level language. As a result, a typical BIOS S3 resume path includes tightly managed assembly code forced to execute from flash memory in order to avoid using system memory. Consequently, this approach has several disadvantages including: (1) an S3 resume implementation is resource limited because the amount of flash memory available for executing S3 resume code is limited; (2) use of the assembly low level language for S3 resume code does not allow for quick portability to a new architecture because assembly code cannot be simply recompiled by a compiler; and (3) the resume codes written in low level assembly language is not modular and less readable to engineers and/or programmers who need to maintain or enhance these codes.

Detailed Description Text (3):
In the discussion below, the teachings of the present invention are utilized to implement a method, apparatus, system, and machine-readable medium for transitioning a computer system from an ACPI S3 sleeping state to a higher power ACPI state. In one embodiment, a portion of the computer system memory is reserved by the system BIOS during a system boot. The reserved portion is used to store a set of instructions to be executed in order to transition the computer system from the ACPI S3 sleeping state to a higher power ACPI state (this set of instructions is also referred to as the S3 resume code herein). The reserved portion is made unavailable for use by the operating system. In one embodiment, in response to a wake up event, the S3 resume code is loaded from flash memory into the reserved portion of system memory. The S3 resume code stored in the reserved portion of the system memory is then executed to transition the computer system from the ACPI S3 sleeping state to a higher power state. In one embodiment, the system BIOS is configured to set a side a predetermined portion of system memory for storing the S3 resume code during the system boot process. The system BIOS is configured to indicate that an area of the system memory available for use by the operating system does not include the reserved portion. In one embodiment, the system BIOS maintains a system memory map to be accessed by the operating system for memory configuration information. The system memory map maintained by the BIOS tells the operating system which area of the system memory is available for use by the operating system which does not include the reserved portion. As a result, the system BIOS can utilize the reserved portion of system memory for execution of S3 resume code without corrupting the OS image stored in a different area of system

memory. The teachings of the present invention are applicable to any scheme, method and system for power management and configuration in computer systems according to ACPI specification. However, the present invention is not limited to ACPI specification and can be applied to the power management and configuration in computer systems with respect to other specifications.

Detailed Description Text (9):
As described above, during a resume from an S3 sleeping state, the system BIOS is responsible for reinitializing the system and transferring control to the OS waking vector without corrupting the OS image. This places an awkward requirement on system BIOS, which is to execute code without using the system memory. This requirement prevents the use of a stack, which is needed in order to develop resume code with a high level language. As a result, a typical BIOS S3 resume path includes tightly managed assembly code forced to execute from flash memory in order to avoid using system memory. Consequently, this approach has several disadvantages including: (1) an S3 resume implementation is resource limited because the amount of flash memory available for executing S3 resume code is limited; (2) use of the assembly low level language for S3 resume code does not allow for quick portability to a new architecture because assembly code cannot be simply recompiled by a compiler; and (3) the resume codes written in low level assembly language is not modular and less readable to engineers and/or programmers who need to maintain or enhance these codes.

Detailed Description Text (10):
To solve the problem described above, the present invention provides a method and a mechanism to allow the system BIOS to reserve and utilize a portion of the system memory for execution of the S3 resume code when the system wakes from an S3 sleeping state. This reserved portion of the system memory is made unavailable for use by the operating system. Thus, the system BIOS is free to use this reserved portion for execution of the S3 resume code without corrupting the OS image. In one embodiment, in order to make the reserved portion unavailable for use by the operating system, the system BIOS after configuring the system memory during the system boot process maintains a system memory configuration map which indicates that the system memory area that is available for use by the operating system does not include the reserved portion for the S3 resume code. In one embodiment, in order to make the operating system unaware of the reserved portion, the system BIOS modifies the top of memory information reported to the operating system via the following INT15 calls: INT15-88, INT15-E801, INT15-E820. These INT functions/subroutines are well-defined runtime functions and are well known in the art. They are used by the operating system to obtain memory configuration information. In ACPI environment, the INT15 interface is required by the ACPI specification to provide the operating system with a system memory map. Specifically, the INT15-88 function returns the size of extended memory (memory above 1 MB). The INT15-E801 function returns the amount of currently installed memory. Specifically, this function, when invoked, will return information with respect to the memory installed between 1 MB and 16 MB, memory installed above 16 MB, memory configured between 1 MB and 16 MB, and memory configured above 16 MB. The function INT15-E820 returns a memory map of all installed RAM and all physical memory ranges reserved by the BIOS. The information returned by this function supersedes the information obtained from the INT15-88 (Return Size of Extended Memory) or the INT15-E801 (Get Memory Size). By modifying the top of memory information reported to the operating system via the INT15 interface, the operating system is unaware of the reserved portion and will leave this area unmodified. The system BIOS then can utilize this reserved area during an S3 resume for code execution without corrupting the OS image.

Detailed Description Text (11):
FIGS. 4A and 4B show an example of a memory map according to the teachings of the present invention. As shown in these figures, one area of the system memory is reserved by the BIOS during the system boot process. This reserved area is utilized

by the system BIOS during an S3 resume for code execution without corrupting the OS image. The size of the reserved area is based upon the size of the S3 resume code and may vary depending upon various implementations. The TOM-BIOS field is the top of memory field that is used by the system BIOS whereas the TOM-OS field is the top of memory field reported to the operating system. If TOM-OS is reported to the operating system instead of TOM-BIOS via the well-defined INT15 interface, the operating system is not aware of the reserved memory area and will leave this area unmodified. The OS image will be loaded in the specified memory having the TOM-OS as the upper boundary. Accordingly, during an S3 resume, the system BIOS can load the S3 resume code from flash memory into the reserved area for execution without disturbing the OS image stored in a different memory area. There are several advantages provided by the present invention. First of all, based on the utilization of a reserved area in system memory for S3 code execution, S3 resume implementation is no longer resource limited because memory resources in modem systems are readily available. This will allow for S3 code generation using a high level language and utilization of memory management schemes such as paging. In addition, the use of high level language for resume code allows for quick portability to a new architecture or upgrades. Resume code written in high level languages can be recompiled by a compiler, as opposed to being rewritten by developers when porting from one architecture to another (e.g., IA-32 to IA-64). Furthermore, resume code written in a high level language is much more maintainable because the resulting source code is modular and more readable.

Detailed Description Text (13):
FIG. 6 shows a flow diagram of one embodiment of a method according to the teachings of the present invention. The system boot begins at block 605. At block 609, the system is initialized, the system memory is configured and a portion of the system memory is reserved for storing S3 resume code. At block 613, a system map containing system memory configuration information is generated. At block 617, the OS boot begins. At block 621, the OS utilizes the memory reporting functions (i.e., the INT15 interface) to obtain memory configuration information. At block 625, the OS loads its image into its corresponding area of system memory which does not include the reserved portion for the S3 resume code. At block 629, the system goes into an S3 sleeping state at some point in time (time X). At block 633, a wakeup event occurs at another point in time (time Y). At block 637, in response to the wakeup event, the BIOS loads the S3 resume code from flash memory into the S3 reserved portion in system memory. At block 641, the S3 resume code is executed to resume the system from the S3 sleeping state. At block 645, the BIOS transfers control to the OS waking vector.

Current US Original Classification (1):
713/300

Current US Cross Reference Classification (2):
713/1

Previous Doc Next Doc Go to Doc#

L29: Entry 3 of 22 File: USPT Jan 20, 2004

DOCUMENT-IDENTIFIER: US 6681336 B1
TITLE: System and method for implementing a user specified processing speed in a computer system and for overriding the user specified processing speed during a startup and shutdown process

Detailed Description Text (32):
The BIOS-ROM 17 stores a system BIOS (Basic Input/Output System), and is constituted by a flash memory so as to rewrite a program. The system BIOS operates in the real mode. This system BIOS includes a POST (Power-On Self Test) routine executed in powering on or restarting the system, a device driver for controlling various I/O devices, a BIOS setup routine for setting the system environment, and a system management program (run time) for executing various SMI processes.

Detailed Description Text (42):
If a system start event (power-on operation, reset, restart, or the like) occurs, POST processing (hardware check and initialization processing) is executed by the system BIOS, and start processing (OS bootstrap sequence) of the operating system (OS) starts. At this time, the CPU 11 operates at a predetermined default speed. In the first embodiment, the highest speed is selected as the default speed of the CPU 11, and the CPU 11 automatically operates at the highest speed until the power-saving driver performs setting processing.

Detailed Description Text (72):
When the system is powered off by suspend processing, the user performs power-on operation to execute resume processing. Resume processing executes processing of restoring information saved in suspend processing to an original location, restoration processing of each software, and resetting processing of each device. Accordingly, the operation state immediately before the start of suspend processing is restored. This resume processing is executed by the OS or in cooperation with the OS and BIOS.

Current US Cross Reference Classification (1):
713/1

L29: Entry 6 of 22 File: USPT May 27, 2003

DOCUMENT-IDENTIFIER: US 6571333 B1
TITLE: Initializing a memory controller by executing software in second memory to wakeup a system

Detailed Description Text (18):
FIG. 4 shows an embodiment using the S2 state. The operating system desires to enter the sleep state in step 410 and stores the resume address used by the BIOS in the RDRAM. The operating system flushes the cache in Step 420, identifies the sleep state by writing the S2 state into the sleep type register, and enables the sleep state by writing the appropriate information into the sleep enable register. The processor and RDRAM clocks are powered down in step 430. In the S2 state, the power to processor 10 is actually removed so that processor 10 is not consuming either active or leakage power.

Detailed Description Text (19):
The system is in the S2 state in step 440. A wake event trigger is detected in step 450. Power is restored to the clocks. A processor reset (CPURST#) is also asserted resetting the processor. The system comes out of reset in step 460 and starts executing software at location FFFFFFF0h. The PAM registers are configured to point to the BIOS storage device and not to shadow this space in the RDRAM.
Alternatively, a hardware state machine can respond to the wake event by changing the PAM registers to point to the BIOS storage device. In step 470, the BIOS initializes the RDRAM. In step 480, the BIOS redirects the PAM registers to execute software from the RDRAM. BIOS passes control to the operating system via the resume address stored in RDRAM in step 410. In step 490, the operating system processes the wake event interrupt. In step 495, recovery from the sleep state is complete and normal operation in the awake state resumes.

Current US Original Classification (1):
713/2

Current US Cross Reference Classification (1):
713/1

Current US Cross Reference Classification (2):
713/100

CLAIMS:

18. The method of claim 17 further comprising: storing a BIOS resume address in the first memory; and transferring control from the BIOS to the operating system following the initializing using the resume address stored in the first memory.

22. The system of claim 21 wherein: the operating system stores a BIOS resume address in the first memory prior to a transition from the awake state to the sleep state; and the BIOS software returns control to the operating system using the stored BIOS resume address after the memory controller is initialized.

L29: Entry 7 of 22 File: USPT Jul 23, 2002

DOCUMENT-IDENTIFIER: US 6425040 B1
** See image for Certificate of Correction **
TITLE: LAN docker unlocking system

Detailed Description Text (67):
The system BIOS reads out the ID (Dock_ID) unique to the LAN docker 200 from its EEPROM 32 to check if the current combination of the PC main body 100 and LAN docker 200 is an ID-registered one, and compares the readout Dock_ID with Dock_ID registered in the flash BIOS-ROM 19. If the two IDs do not match, the system BIOS turns off the power supply of the PC main body 100, and resets its system status to a state (suspend/hibernation/stop state) before generation of the wake-up signal Wake_up. On the other hand, if the two IDs match, the system BIOS turns on the Q switch 16, executes a resume process from the suspend/hibernation state or starts up the OS, and passes the control to the OS. In this fashion, the PC main body 100 can be accessed from the server via the network using the LAN controller 31.

Detailed Description Text (74):
If Dock_ID_FLAG="0", the system BIOS determines that the WOL function of the PC main body 100 is not enabled and does not correspond to any LAN docker. The system BIOS refers, in turn, to the wake-on-LAN flag (WOL_FLAG) of the currently attached LAN docker 200 to check if that LAN docker 200 corresponds to another PC to enable the WOL function (steps S122 and S123). If WOL_FLAG="0", i.e., if the WOL function is not enabled and that LAN docker does not correspond to any PC, the system BIOS determines that the WOL functions of the current combination of the PC main body 100 and LAN docker 200 are not enabled and this combination is not an ID-registered one. Then, the system BIOS starts the above-mentioned docking process including processes for turning on the Q switch 16, canceling the reset state, and the like (step S128), so that the LAN controller 31 can be used. With this process, the LAN controller 31 is set in the operative state. After that, the system BIOS executes a system startup process such as a resume process from the suspend/hibernation state, an OS startup process, or the like to set the system state in the operative state, and passes control to the OS (step S130). The PC main body 100 is used in the locked state.

Current US Cross Reference Classification (2):
713/300

L29: Entry 9 of 22 File: USPT Apr 2, 2002

DOCUMENT-IDENTIFIER: US 6367074 B1
TITLE: Operation of a system

Detailed Description Text (8):
As examples, the off states may include states defined under the Advanced Configuration and Power Interface (ACPI) Specification, Revision 1.0, dated Dec. 22, 1996. Example off states as defined by the ACPI specification include a mechanical off state (in which power is cut off from components in the system), a soft off state, and several sleep states. In some of the defined sleep states, system context is lost. To resume from one of these states to a working state, a BIOS routine restores some settings from a non-volatile storage medium and control is passed to the operating system, which may resume executing from an address location stored before the system entered the sleep state. Other off states may include those defined by the Advanced Power Management (APM) BIOS Interface Specification, Version 2.1, dated February 1996.

Detailed Description Text (26):
If a full-feature operating system boot is desired (as determined at 212 or 206), then the BIOS routine 404 proceeds (at 216) to either boot or resume the operating system 400, depending on which off state the system 10 was in. The operating system 400 is booted if the system was initially in the powered off state. The operating system 400 is resumed if the system was originally in a low power state in which system context information was saved to a non-volatile storage medium.

Current US Cross Reference Classification (1):
713/2

L29: Entry 10 of 22 File: USPT May 29, 2001

DOCUMENT-IDENTIFIER: US 6240530 B1
** See image for Certificate of Correction **
TITLE: Virus extermination method, information processing apparatus and computer-readable recording medium with virus extermination program recorded thereon

Detailed Description Text (61):
The information processing apparatus 1 further includes a controller 14 which clears, when the ROM 4-2 is effectively connected to the bus 13 via the gate 9-2, the RAM 3 based on the BIOS stored in the ROM 4-2 to stop the resume function, and thereafter loads the OS and so forth into the RAM 3 based on the IPL or boot stored in the storage apparatus 12 so that system environment which is not infected with a virus can be established.

Detailed Description Text (172):
When the connection state of the gate 9-2 is established, the ROM 4-2 is put into an access enable state. Thus, based on the BIOS stored in the ROM 4-2, the CPU 2 stops the resume function by deleting all of the stored contents of the RAM 3 as an object with which resume control is proceeding, and then starts up the OS 12a stored in the storage apparatus 121 and free from virus infection. In other words, after the stored information of the RAM 3 and so forth is cleared, an operating system in which no virus is resident is fetched from the outside and started up.

Current US Cross Reference Classification (1):
713/1

L29: Entry 11 of 22 File: USPT Jul 11, 2000

DOCUMENT-IDENTIFIER: US 6088794 A
TITLE: Computer system capable of selective booting from two hard disk drives

Brief Summary Text (20):
According to this invention, since boot hard disk drives selection and suspend/resuming operation for two hard disk drives is possible by user interaction in a complementary metal oxide semiconductor (CMOS) setup program or a basic input output system (BIOS) program, dual operating system can be used in a computer system in an easy and efficient way. Further, the present invention provides a remedy for the hard disk failure in the event that the boot sector of the master hard disk drive has failed, if the first and second hard disk drives store the same operating system.

Detailed Description Text (35):
As apparent from the foregoing description, this invention provides a dual bootable computer in which selective booting as well as suspend/resuming operation can be performed with respect to both first and second hard disk drives. Also, since boot hard disk drives selection and suspend/resuming operation for two hard disk drives is possible by user interaction in a complementary metal oxide semiconductor (CMOS) setup program or a basic input output system (BIOS) program, dual operating system can be used in a computer system in an easy and efficient way.

Current US Original Classification (1):
713/2

L29: Entry 12 of 22 File: USPT Oct 26, 1999

DOCUMENT-IDENTIFIER: US 5974554 A
TITLE: Computer system with automatic configuration capability for industry standard architecture (ISA) cards

Brief Summary Text (18):
Typically, when a computer system using an advanced power is booted, either a resume boot process or a BIOS boot process is performed depending upon whether the system is in a resume mode. If the BIOS boot process is performed, an operating system is loaded and executed. After the operating system is executed or after the resume boot process is performed, the system enters a normal mode of operation.

Brief Summary Text (19):
For many PC systems using PnP incompatible operating systems (e.g., MS-DOS.RTM. and OS/2.RTM.) which do not support the PnP automatic configuration mechanism, the system BIOS may directly save and restore the PnP card information during suspend and resume modes. However, unlike PC systems using PnP incompatible operating systems, PC systems having PnP compatible operating system (e.g., WINDOWS 95.RTM., or the like) and PnP ISA device cards, and employing such a power management scheme experience significant problems when the PC system enters into a suspend mode. The configuration of PnP ISA device cards is cleared when the PC system is powered off in a suspend mode. During a resume mode, the information of each PnP ISA device card needs restoring when the corresponding device driver receives the resume message from the BIOS through the operating system. However, most of the device drivers do not support this function. Further, in such a PnP compatible PC system, the system BIOS cannot find the address of the READ.sub.-- DATA port since the PnP compatible operating system (specifically, WINDOWS 95.RTM.) directly configures the READ.sub.-- DATA port of PnP ISA cards and does not re-configure the PnP cards. Thus, it is impossible for the system BIOS to save/restore the PnP ISA card information when the PC system is entering the suspend/resume mode, thereby disabling the PnP ISA cards after the system resume.

Brief Summary Text (23):
These and other objects of the present invention can be achieved by a computer system with an advanced power management comprises a Plug and Play (PnP) compatible operating system; an industry standard architecture (ISA) bus; a non-volatile storage unit; at least one PnP ISA expansion device coupled to the ISA bus; and a basic input/output operating system (BIOS) for searching a read data port for the PnP expansion device to save/restore information of the PnP ISA expansion device into/from the non-volatile storage unit during a suspend/resume mode of the advanced power management.

Detailed Description Text (2):
Referring now to the drawings and particularly to FIG. 1, which illustrates a booting process of a typical computer system using an advanced power management. Referring to FIG. 1, from the start at step S10, it is checked whether the system will enter a resume mode at step S20. If the system is in a resume mode, the resume boot process is performed at step S30. If the system is not in a resume mode, the BIOS boot process is performed at step S40, and then an operating system is loaded and executed at step S50. After either step S30 or step S50, the system is in a normal mode of operation at step S60.

Detailed Description Text (3):
However, unlike PC systems using PnP incompatible operating systems, PC systems having PnP compatible operating system (e.g., WINDOWS 95.RTM., or the like) and PnP ISA device cards, and employing such a power management scheme experience significant problems when the PC system enters into a suspend mode. The configuration of PnP ISA device cards is cleared when the PC system is powered off in a suspend mode. During a resume mode, the information of each PnP ISA device card needs restoring when the corresponding device driver receives the resume message from the BIOS through the operating system. However, most of the device drivers do not support this function. Further, in such a PnP compatible PC system, the system BIOS cannot find the address of the READ.sub.-- DATA port since the PnP compatible operating system (specifically, WINDOWS 95.RTM.) directly configures the READ.sub.-- DATA port of PnP ISA cards and does not re-configure the PnP cards. Thus, it is impossible for the system BIOS to save/restore the PnP ISA card information when the PC system is entering the suspend/resume mode, thereby disabling the PnP ISA cards after the system resume.

Current US Original Classification (1):
713/300

CLAIMS:

1. A computer system with an advanced power management, comprising:

an industry standard architecture (ISA) bus;

a non-volatile memory;

at least one PnP ISA expansion device coupled to the ISA bus; and

a basic input/output operating system (BIOS) for searching a read data port for said PnP expansion device to save information of said PnP ISA expansion device into said non-volatile memory during a suspend mode of the advanced power management, and to restore information of said PnP ISA expansion device from said non-volatile memory during a resume mode of the advanced power management.

Previous Doc Next Doc Go to Doc#

L29: Entry 19 of 22 File: USPT Nov 19, 1996

DOCUMENT-IDENTIFIER: US 5577220 A
TITLE: Method for saving and restoring the state of a CPU executing code in protected mode including estimating the value of the page table base register

Detailed Description Text (105):
It is believed that although any discussion of the computer system 10 of the present invention is somewhat circular because most of the routines interract with the others and the suspend/resume process is a continuing cycle, a discussion of the Suspend Routine (FIG. 10) before the Boot Routine (FIG. 11) or the Resume Routine (FIG. 12) will be most helpful. Referring now to FIG. 10, a flow chart of the Suspend Routine is shown. Recall that after either the normal boot routine 204-210 or the resume boot routine 214-220 are executed, the computer system 10 is in the normal operating state 150. Moreover, as mentioned above in the text accompanying FIG. 8, whether the computer system was either normally booted 204-210 or resume-booted 214-220, after either routine finishes, the APM OS driver is aware of the APM BIOS routines, such as the Supervisor Routine, shown in FIG. 8. As a result, the APM polls the Supervisor Routine approximately every one second.

Current US Cross Reference Classification (2):
713/300

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#
Generate Collection Print

L29: Entry 21 of 22 File: USPT Oct 11, 1994

DOCUMENT-IDENTIFIER: US 5355490 A
TITLE: System and method for saving the state for advanced microprocessor operating modes

Brief Summary Text (9):
Unfortunately, the resume processing BIOS code presently built into such laptops supports the resume function under the basic MS-DOS operating system only while the microprocessor is operating in the "real mode". When run under a different operating environment, such as the Windows operating environment which is a graphical user interface available from Microsoft Corporation of Redmond, Wash., the present BIOS resume function does not properly save certain registers and other data used by the microprocessor, and the resume function operation thereby fails. This is due, in part, to the fact that the Windows operating environment was designed to operate on a computer system utilizing a multi-mode 80386 microprocessor operating in an advanced mode, such as the "enhanced mode", or a microprocessor downwardly compatible with the 80386, such as the 80486, etc. On the other hand, the MS-DOS operating system was designed to operate on an 8086/8088 microprocessor utilizing only one "real" mode.

Current US Original Classification (1):
713/100

Current US Cross Reference Classification (2):
713/2

Previous Doc Next Doc Go to Doc#

L29: Entry 22 of 22 File: USPT Aug 16, 1994

DOCUMENT-IDENTIFIER: US 5339426 A
**** See image for Certificate of Correction ****
TITLE: System and method for resume processing initialization

Brief Summary Text (9):
Unfortunately, the resume processing BIOS code presently built into such laptops supports the resume function under the MS-DOS operating system only. When run under a different operating system, such as OS/2 which is a multi-tasking operating system available from Microsoft Corporation of Redmond, Wash., the present BIOS resume function does not properly save certain registers and other data used by OS/2, and the resume function operation thereby fails. This is due, in part, to the fact that the OS/2 operating system was designed to operate on a computer system utilizing a multi-mode 80286 microprocessor or a microprocessor downwardly compatible with the 80286, such as the 80386, 80486, etc. On the other hand, the MS-DOS operating system was designed to operate on an 8086/8088 microprocessor utilizing only one mode.

Brief Summary Text (13):
It is a further objective of the present invention to provide a resume processing function for use with OS/2 in conjunction with an existing resume processing routine, such as the MS-DOS resume processing BIOS routine.

Detailed Description Text (16):
In a preferred embodiment, the implementation of the present invention involves (1) creating a resume driver capable of operating under the OS/2 Version 1.2 operating system in order to hook the NMI interrupt vector as well as to act as a wrapper around the Resume BIOS code, (2) modifying the system loader in order to insert a hook into the real mode non-maskable interrupt (NMI), thereby allowing the present invention to operate under OS/2's DOS compatibility Box, (3) modifying the system initialization process to place the resume driver load statement in the "CONFIG.SYS" file when installing on an appropriate machine, and (4) modifying the disk driver to signal to the Resume BIOS when disk activity has subsided so that the power may be removed without losing disk buffers or data. As will be explained in the detailed discussion below, all of these components will have to be written or modified in order to fully support the resume function under OS/2 Version 1.2, but it will be readily understood by one of ordinary skill in the art that the described modifications and additions will define only one implementation of the present claimed invention, and that other obvious variations will be apparent from the discussion.

Detailed Description Text (37):
Referring again to step 202, the resume processing driver of the present invention is executed after the NMI is detected by the CPU and the NMI is determined to be a result of the power-down switch being pressed. At this point, the resume driver performs various steps in order to properly save certain types of data associated with OS/2 into appropriate areas in the computer system's memory. After this is performed, the resume driver then transfers control to pre-existing ROM BIOS resume code which performs further resume processing functions and then physically removes

power supplied to the various components of the computer system, except the memory. Thus, the resume driver of the present invention, depicted in step 202, serves as a "wrapper" around the pre-existing ROM BIOS currently found in various computer systems, such as Toshiba laptops. That is, the resume driver saves the registers and data unique to the OS/2 operating system and/or the advanced microprocessor upon which OS/2 operates and places the computer system into a state in which the standard ROM BIOS resume processing routine is able to properly perform its functionality and thereafter remove power from the various components.

Detailed Description Text (44):
Referring to FIG. 3, a diagram is shown which illustrates the steps necessary in order to "hook" the real mode NMI vector. Basically, the OS/2 system loader 300 must be modified in order to perform the following functions. First, the loader must save the existing NMI vector which points to the standard resume ROM BIOS. Because the actual NMI vector in the IVT (element 301) cannot be modified because the OS/2 kernel polices any changes made, the first step of the original handler should be changed to jump to a far address corresponding to the resume driver of the present invention. However, because the original handler resides in the ROM BIOS (and hence may not be modified), the next best solution is to modify the IVT entry for the real mode NMI in the loader itself. This will fool the kernel into thinking that this new vector is the proper vector. Unfortunately, at load time, there is no way of determining where the resume driver will eventually be placed into memory, so the IVT NMI vector (element 301) is simply pointed to a dummy handler within the loader itself, as depicted in FIG. 3. The dummy handler will consist of five NOP (no-operation) instructions followed by a far jump to the BIOS NMI handler, as depicted in elements 302A and 304 of FIG. 3.

Detailed Description Text (50):
In addition to the modifications to the OS/2 system loader as described above, the OS/2 disk device driver must be modified in order to fully support the resume functionality of the present invention. This is necessary because the standard resume routines found in the ROM BIOS expects certain registers to be modified by the disk BIOS. However, under the protected mode of OS/2 Version 1.2, the disk BIOS code is not used. Accordingly, modifications must be made to the disk driver in order to perform the steps described below. The modifications to the disk driver according to the preferred embodiment described below are relatively simple in scope, and one of ordinary skill will be able to make these modifications to the OS/2 disk driver code.

Detailed Description Text (59):
The resume driver initialization routine performs various functions. First, step 502 determines whether the host computer system is a type of computer system which supports the resume function of the present invention. This determination may be made by reading the appropriate memory locations containing codes specifying which type of computer system is present (e.g. ROM BIOS address F000:FFFA for the Toshiba line of computers), and by reading the appropriate memory locations indicating whether the standard ROM BIOS resume function is supported (e.g. ROM BIOS address F000:E024, et seq., on Toshiba computers). Specifically, the computer system must be able to operate under the OS/2 operating system (for example, a computer system with an 80268-compatible microprocessor--CPU 101 in FIG. 1), it must have an intelligent power supply circuit (element 117 in FIG. 1), and it must have an existing resume processing functionality with its ROM BIOS. In a preferred embodiment, either the Toshiba T3100SX or T2000SX laptop computers satisfy this criteria, and therefore step 502 makes specific reference to these machines. However, any other computer system operating under OS/2 with an intelligent power supply may be used to implement the present invention, as well, and the T3100SX and the T2000SX are mentioned for illustrative purposes only.

Detailed Description Text (75):
After both the real mode and protected mode NMI vectors have been "hooked" in steps

603 and 604, step 605 is encountered, and the timer handler is exited. Once this step has been completed, the OS/2 resume driver of the present invention has been installed, and may be utilized when the user of the computer system presses the power switch, thereby sending an NMI to the CPU. The following discussion describes in detail the various steps which constitute the main routine of the resume driver. Accordingly, these steps are performed during the actual OS/2 resume processing and before the standard resume routines located with the ROM BIOS are executed.

Detailed Description Text (77):
In the event of an I/O channel check error NMI (parity error), which in the preferred embodiment is the only other type NMI which can occur other than a resume NMI, the resume driver of the present invention performs preprocessing and transfers control to the BIOS, which halts the system. At this point the machine will be halted and the user will have to press the reset button to re-boot the system. The actual execution path followed by the resume driver is the same regardless of the cause of the NMI.

Detailed Description Text (97):
The CMOS real time clock settings must be specifically reset for OS/2 by the resume driver, as depicted by step 715, because the BIOS code does not perform this function. For example, on the T3100SX computer: (1) the time-base frequency is set to 32,768 KHz, (2) the rate selection bits are set so that the interrupt rate is every 31.25 ms, and (3) periodic interrupts, update ended interrupts, and 24-hour mode interpretation are turned on. In this case, appropriate instructions would be executed to achieve these effects.

Current US Original Classification (1):
713/1

Previous Doc Next Doc Go to Doc#